I. Configure the project in Apache Maven

For configuring a java project in Apache Maven, see this article.
In this project, Maven is used to build the java source code by configuring all the dependencies and create a .jar executable binary of it the target/ folder.

1. Create an Apache Maven project hierarchy using following command on the command line

```
mvn archetype:generate -DgroupId=minor -DartifactId=devops_minor -
DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -
DinteractiveMode=false
```

- Group ID: It is the package name of our project. It is kept as **minor** here.
- Artifact ID: Is is the name for our Maven project. It is kept as **devops_minor** here.
- All other tags set up the maven hierarchy automatically.

```
500069750@upes-500069750 MINGW64 ~
$ mvn archetype:generate -DgroupId=minor -DartifactId=devops_minor -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------< org.apache.maven:standalone-pom >-------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] --------------------------------[ pom ]---------------------------------
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.0:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.2.0:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO]
[INFO]
[INFO] --- maven-archetype-plugin:3.2.0:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: minor
[INFO] Parameter: artifactId, Value: devops_minor
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: minor
[INFO] Parameter: packageInPathFormat, Value: minor
[INFO] Parameter: package, Value: minor
[INFO] Parameter: groupId, Value: minor
[INFO] Parameter: artifactId, Value: devops_minor
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\Users\500069750\devops_minor
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:17 min
[INFO] Finished at: 2021-04-29T10:55:59+05:30
[INFO] ------------------------------------------------------------------------

500069750@upes-500069750 MINGW64 ~
$
```

2. Copy our java code in the Maven hierarchy

- Go to **src/main/java/{package name}/** and delete the template **App.java** file. Here, package name is minor.
- Copy our **Main.java** code in the current directory.
- Modify the **Main.java** code by placing the following line on the top of the code.

```
import minor.*;
```

3. Clean the project hierarchy using Maven on command line

```
$ mvn clean
```

Clean goal of Maven is used to clear the cache in the Maven hierarchy.

```
500069750@upes-500069750 MINGW64 ~/devops_minor
$ mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< minor:devops_minor >------------------------
[INFO] Building devops_minor 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ devops_minor ---
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.560 s
[INFO] Finished at: 2021-04-29T11:08:48+05:30
[INFO] ------------------------------------------------------------------------

500069750@upes-500069750 MINGW64 ~/devops_minor
$
```

If previous builds are present in the target/ folder, the target/ folder is also deleted for a fresh start.

4. Compile the java source code

```
$ mvn compile
```

```
500069750@upes-500069750 MINGW64 ~/devops_minor
$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< minor:devops_minor >------------------------
[INFO] Building devops_minor 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ devops_minor ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\500069750\devops_minor\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ devops_minor ---
[INFO] Nothing to compile - all classes are up to date
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.397 s
[INFO] Finished at: 2021-04-29T11:09:17+05:30
[INFO] ------------------------------------------------------------------------
```

Maven compile goal is used to compile all the source code files in the **src/main/java/{packagename}** folder

5. Create a build of the project using Install goal

```
$ mvn install
```

Install goal of Maven creates a binary executable .jar file of the project which is stored in the **target/** folder.

```
500069750@upes-500069750 MINGW64 ~/devops_minor
$ mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< minor:devops_minor >------------------------
[INFO] Building devops_minor 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ devops_minor ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\500069750\devops_minor\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ devops_minor ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @ devops_minor ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\500069750\devops_minor\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ devops_minor ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\500069750\devops_minor\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ devops_minor ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running minor.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.044 s - in minor.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ devops_minor ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: C:\Users\500069750\devops_minor\target\devops_minor-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ devops_minor ---
[INFO] Installing C:\Users\500069750\devops_minor\target\devops_minor-1.0-SNAPSHOT.jar to C:\Users\500069750\.m2\repository\minor\devops_minor\1.0-SNAPSHOT\devops_minor-1.0-SNAPSHOT.jar
[INFO] Installing C:\Users\500069750\devops_minor\pom.xml to C:\Users\500069750\.m2\repository\minor\devops_minor\1.0-SNAPSHOT\devops_minor-1.0-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.226 s
[INFO] Finished at: 2021-04-29T11:09:56+05:30
[INFO] ------------------------------------------------------------------------
```

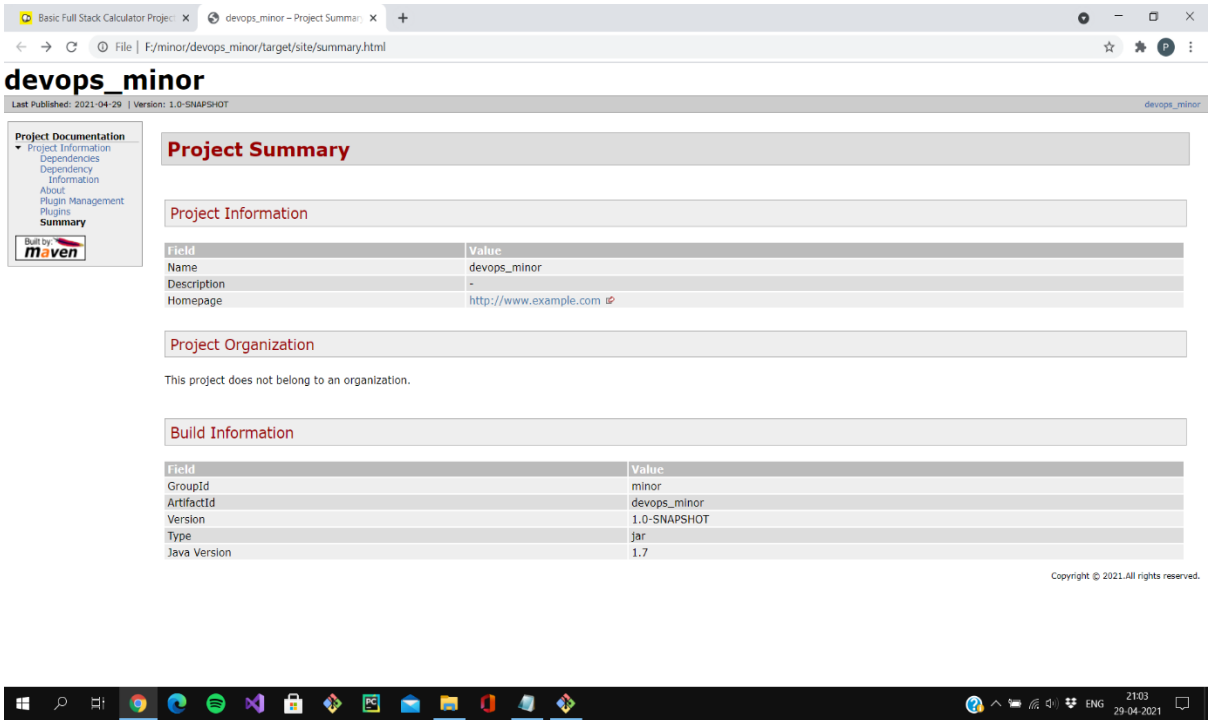
6. Maven Site goal

Maven site goal is used to automatically create a documentation report of the project using HTML and CSS in the **target/site** folder.

```
$ mvn site
```

```
500069750@upes-500069750 MINGW64 ~/devops_minor
$ mvn site
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< minor:devops_minor >------------------------
[INFO] Building devops_minor 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-site-plugin:3.7.1:site (default-site) @ devops_minor ---
[INFO] configuring report plugin org.apache.maven.plugins:maven-project-info-reports-plugin:3.0.0
[INFO] 15 reports detected for maven-project-info-reports-plugin:3.0.0: ci-management, dependencies, dependency-info, dependency-management, distribution-management, index, issue-management, licenses, mailing-l
ists, modules, plugin-management, plugins, scm, summary, team
[INFO] Rendering site with default locale English (en)
[INFO] Relativizing decoration links with respect to localized project URL: http://www.example.com
[INFO] Rendering content with org.apache.maven.skins:maven-default-skin:jar:1.2 skin.
[INFO] Generating "Dependencies" report  --- maven-project-info-reports-plugin:3.0.0:dependencies
[INFO] Generating "Dependency Information" report --- maven-project-info-reports-plugin:3.0.0:dependency-info
[INFO] Generating "About" report         --- maven-project-info-reports-plugin:3.0.0:index
[INFO] Generating "Plugin Management" report --- maven-project-info-reports-plugin:3.0.0:plugin-management
[INFO] Generating "Plugins" report       --- maven-project-info-reports-plugin:3.0.0:plugins
[INFO] Generating "Summary" report       --- maven-project-info-reports-plugin:3.0.0:summary
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.409 s
[INFO] Finished at: 2021-04-29T11:10:26+05:30
[INFO] ------------------------------------------------------------------------
```

The summary of our current Maven project made by the mvn site goal:-

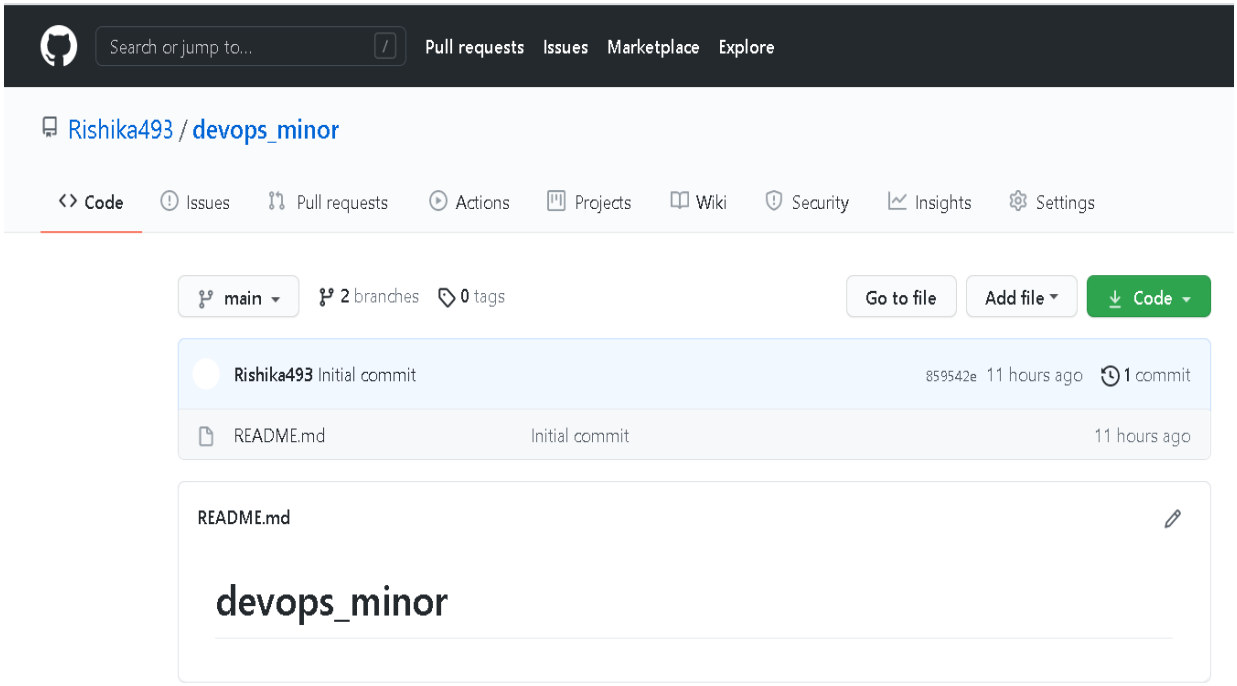


II. Host the project on GitHub

1. Create a new repository on GitHub

Here, we create a repository with the same name as that of the Maven project artifact ID.

Go to your GitHub account -> Your Repositories -> New repository.



2. Configure the Maven project on local machine and link it to the Git repository

Step 1. Initialize the Maven project as a Git repository

```
$ git init
```

Step 2. Link the Git repository with local repository

In this step, we establish a link between the project stored on our local machine and the repository we created on the GitHub account.

Copy the URL of the repository hosted on Git and type the following command on the command line:

- **origin:** It is the name of the link between remote repository and local repository. It is a convention to use origin as the link name but we can use any other name as well.

```
$ git remote add origin "https://github.com/Rishika493/devops_minor"
```

```
[INFO] ------------------------------------------------------------------------

500069750@upes-500069750 MINGW64 ~/devops_minor
$ git init
Initialized empty Git repository in C:/Users/500069750/devops_minor/.git/

500069750@upes-500069750 MINGW64 ~/devops_minor (master)
$ git remote add origin "https://github.com/Rishika493/devops_minor"

500069750@upes-500069750 MINGW64 ~/devops_minor (master)
$ git add .
warning: LF will be replaced by CRLF in target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/createdFiles.lst.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/inputFiles.lst.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in target/site/css/maven-base.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in target/site/css/print.css.
The file will have its original line endings in your working directory
```

3. Stage the Maven project in Git environment

The files which are to be version controlled by Git have to be staged.After a file is once staged, git always checks whether it is modified, deleted, renamed etc.

A file must be staged before it can be pushed to the remote Git repository.

Here, following command is used.

```
$ git add .
```

- . command specifies stage all the files in the current folder that are modified or created.

```
$ git remote add origin "https://github.com/Rishika493/devops_minor"

500069750@upes-500069750 MINGW64 ~/devops_minor (master)
$ git add .
warning: LF will be replaced by CRLF in target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/createdFiles.lst.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/inputFiles.lst.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in target/site/css/maven-base.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in target/site/css/print.css.
The file will have its original line endings in your working directory
```

4. Commit the staged files to GitHub

Staged files can be committed to the remote Git repository using the following command.Committing a file means that the file is mirrored in the Git repository.

```
$ git commit -m "Commit Message"
```

```
500069750@upes-500069750 MINGW64 ~/devops_minor (master)
$ git commit -m "Java Project in Maven configured and build successfully"
[master (root-commit) 30567c2] Java Project in Maven configured and build successfully
 34 files changed, 1650 insertions(+)
 create mode 100644 pom.xml
 create mode 100644 src/main/java/minor/Main.java.txt
 create mode 100644 src/test/java/minor/AppTest.java
 create mode 100644 target/devops_minor-1.0-SNAPSHOT.jar
 create mode 100644 target/maven-archiver/pom.properties
 create mode 100644 target/maven-status/maven-compiler-plugin/compile/default-compile/inputFiles.lst
 create mode 100644 target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/createdFiles.lst
 create mode 100644 target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/inputFiles.lst
 create mode 100644 target/site/css/maven-base.css
 create mode 100644 target/site/css/maven-theme.css
 create mode 100644 target/site/css/print.css
 create mode 100644 target/site/css/site.css
 create mode 100644 target/site/dependencies.html
 create mode 100644 target/site/dependency-info.html
 create mode 100644 target/site/images/close.gif
 create mode 100644 target/site/images/collapsed.gif
 create mode 100644 target/site/images/expanded.gif
 create mode 100644 target/site/images/external.png
 create mode 100644 target/site/images/icon_error_sml.gif
 create mode 100644 target/site/images/icon_info_sml.gif
 create mode 100644 target/site/images/icon_success_sml.gif
 create mode 100644 target/site/images/icon_warning_sml.gif
 create mode 100644 target/site/images/logos/build-by-maven-black.png
 create mode 100644 target/site/images/logos/build-by-maven-white.png
 create mode 100644 target/site/images/logos/maven-feather.png
 create mode 100644 target/site/images/newwindow.png
 create mode 100644 target/site/index.html
 create mode 100644 target/site/plugin-management.html
 create mode 100644 target/site/plugins.html
 create mode 100644 target/site/project-info.html
 create mode 100644 target/site/summary.html
 create mode 100644 target/surefire-reports/TEST-minor.AppTest.xml
 create mode 100644 target/surefire-reports/minor.AppTest.txt
 create mode 100644 target/test-classes/minor/AppTest.class
```

5. Push the changes to the GitHub repository

Git commit sends the file from local repository to the remote repository but they will not be shown on the remote repository unless they are pushed.
Push synchronizes the remote Git repository with the latest committed changes from local repository.

```
$ git push -u origin master
```

Here,

- **origin** is the link name defined in the step 2.
- **master** is the default branch

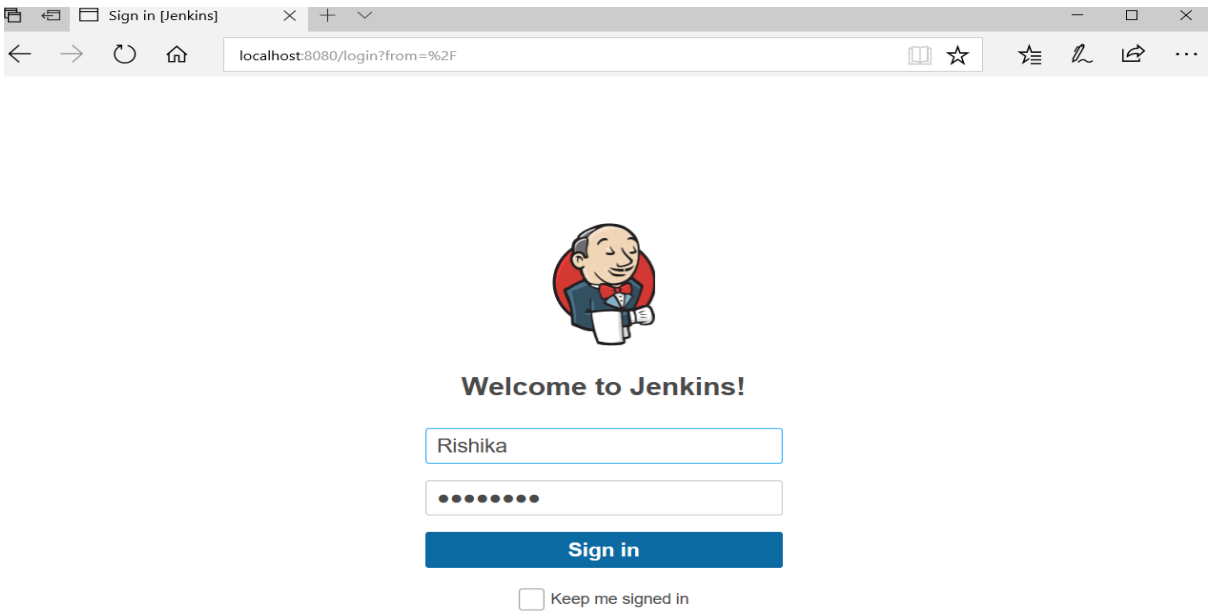
```
500069750@upes-500069750 MINGW64 ~/devops_minor (master)
$ git push -u origin master
Logon failed, use ctrl+c to cancel basic credential prompt.
Enumerating objects: 58, done.
Counting objects: 100% (58/58), done.
Delta compression using up to 8 threads
Compressing objects: 100% (43/43), done.
Writing objects: 100% (58/58), 28.23 KiB | 1.13 MiB/s, done.
Total 58 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/Rishika493/devops_minor/pull/new/master
remote:
To https://github.com/Rishika493/devops_minor
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

500069750@upes-500069750 MINGW64 ~/devops_minor (master)
$ 9~
```

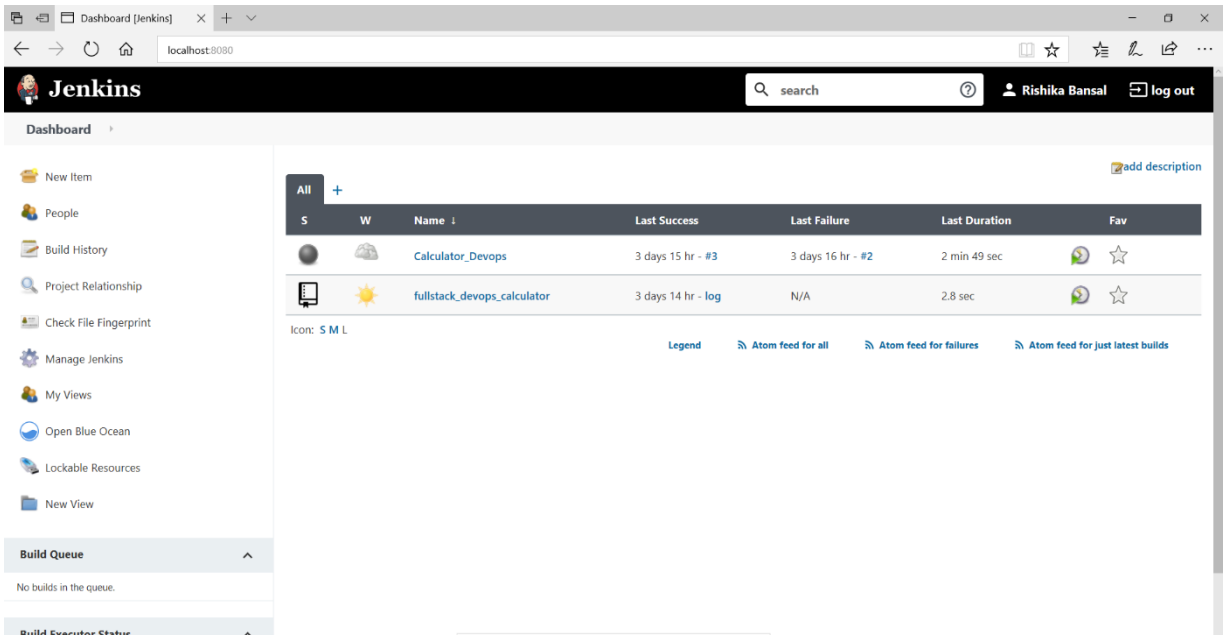
III. Jenkins Setup (Creating pipeline): -

1.For the interface of Jenkins, open the web browser and go to localhost:8080

A signup page loads, create a new user and sign in.

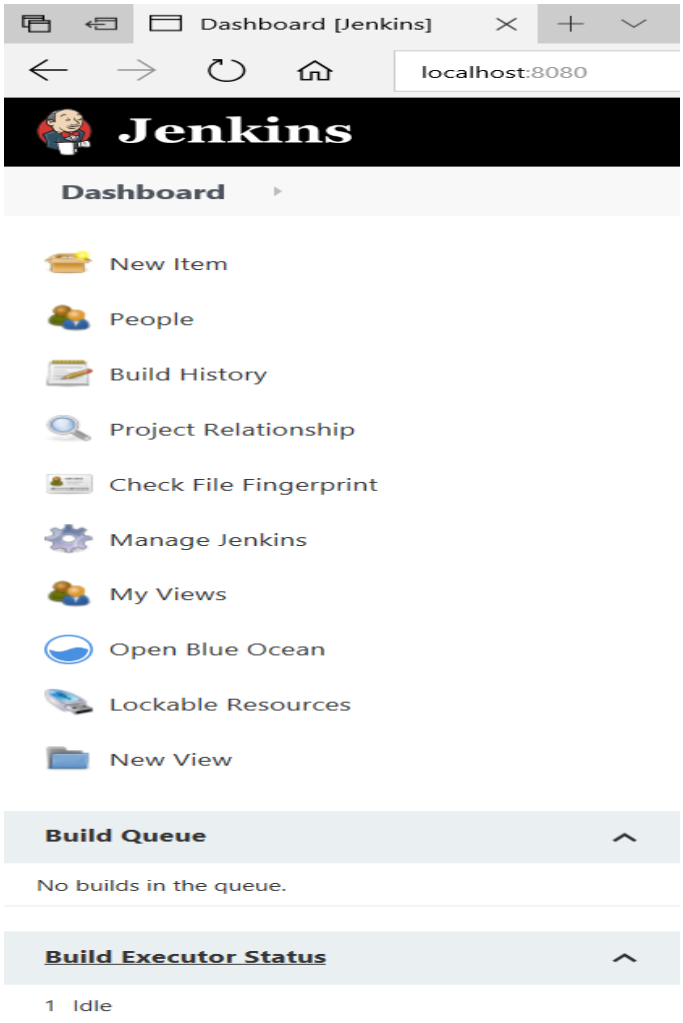


2. Jenkins dashboard opens up



Here, we create a Jenkins job to execute Maven tasks like clean, compile and install on the project hosted on Git.

3. Click on New Item on left sidebar on the Jenkins Dashboard.



4. Enter the project name and select Freestyle Project. Click on OK.

5.Project configuration Windows opens.

In the **Source Code Management** tab, select the **Git** option and specify the URL of the GitHub project.

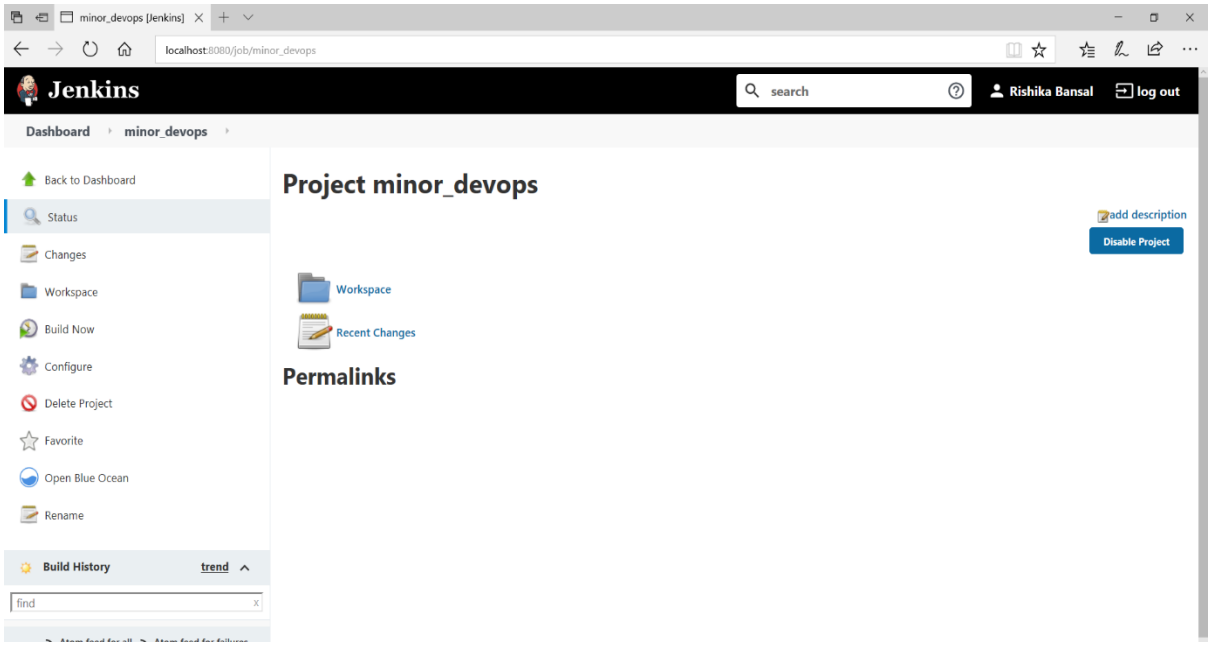
Note: The pom.xml file of our Maven project must be on the root of the GitHub repository.

In the **Build Environment** tab, select **Invoke top level Maven Targets** from Build dropdown menu and specify the following goals in order:

Clean compile install

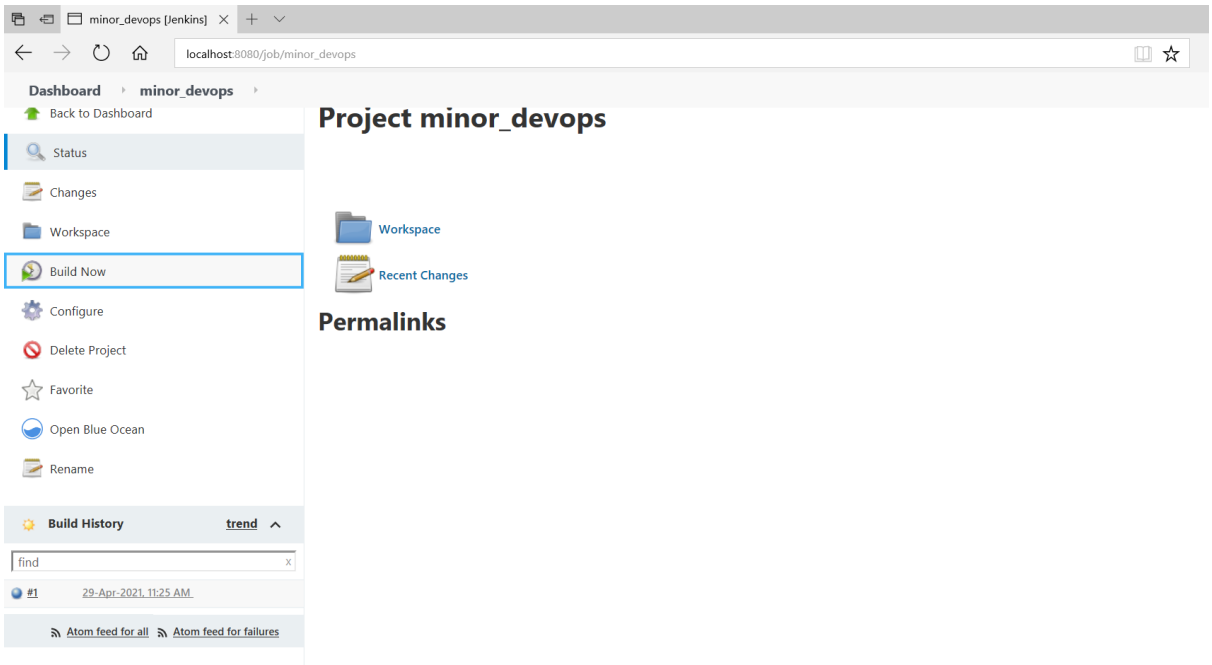
Click on **Save**.

6. The project is created and its dashboard is displayed.

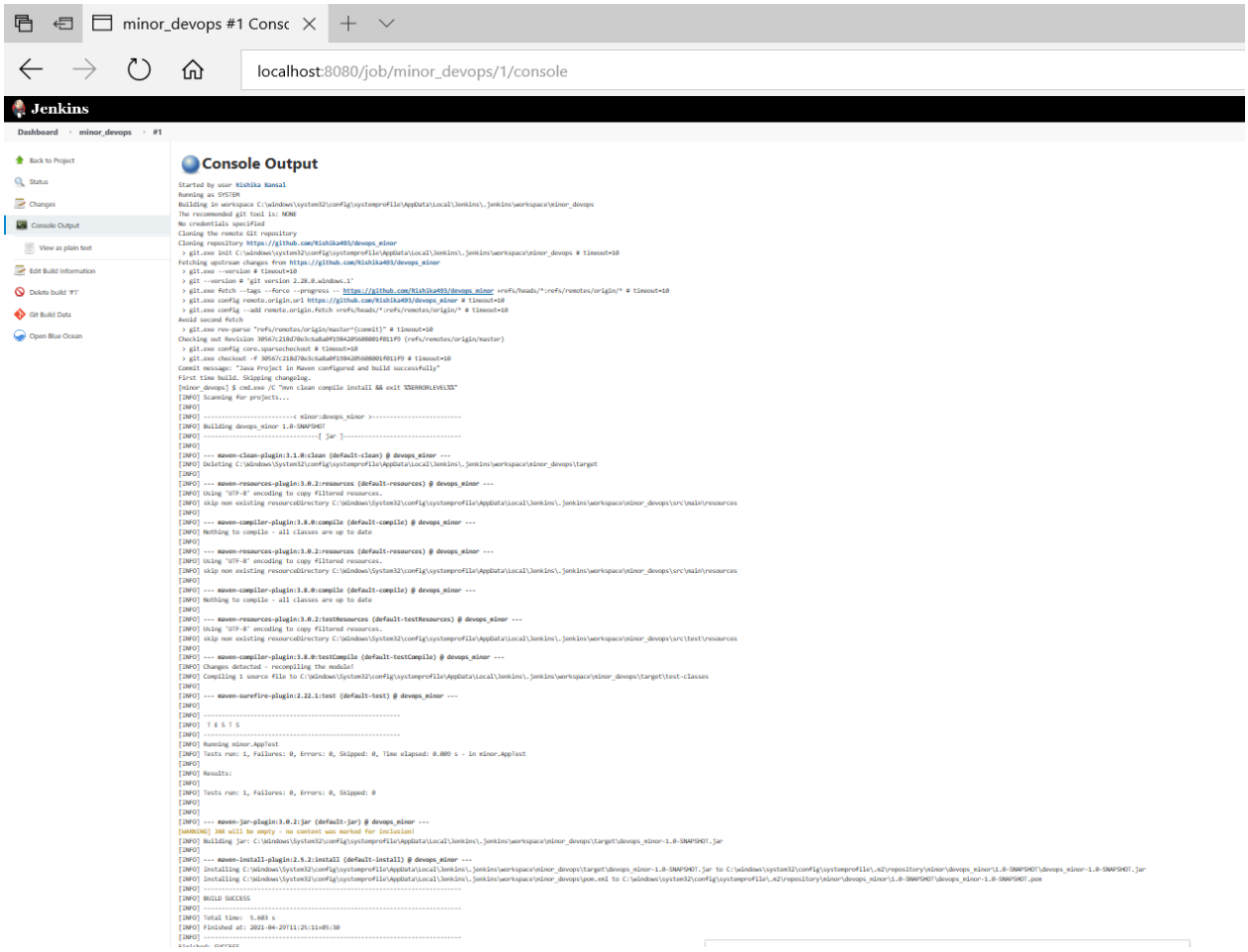


7. Click on Build Now option on the sidebar.

8. After the project is built, the build history shows the Build with its build number on the bottom of the left sidebar.



Click on the **build number (Here, #1)** and click on **Console Output** to see the Build history.



Console output ends with the message **Build Successful** which means our project is successfully built using Jenkins.

9. Creating an automated Pipeline for our project using Jenkins

We have created a Jenkins job to execute Maven goals like clean, compile and install on the project hosted on GitHub. Here, we create a pipeline that shows the progress of each task. For this, we use the BlueOcean pipeline plugin of Jenkins.

9.1 Create a **Jenkinsfile** (same file name and no extension) and save this file on the root of our GitHub repository.

```
pipeline {
  agent any
  stages
    {
    stage('Clean') {
      steps {
        bat 'mvn clean'
      }
    }
    stage('Compile') {
```

```
      steps {
        bat 'mvn compile'
      }
    }
    stage('Test') {
      steps {
        bat 'mvn test'
      }
    }
  }
}
```

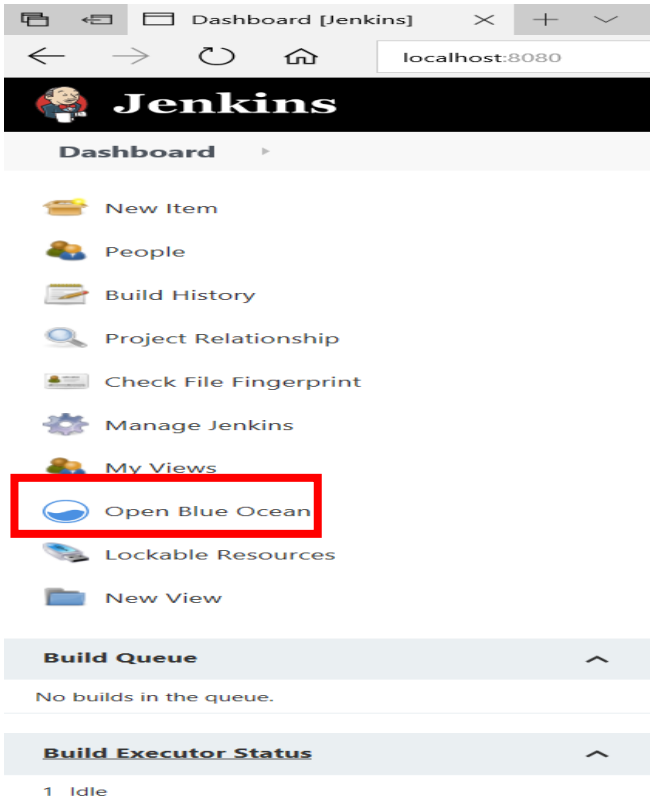
Jenkinsfile

Copy

- **Stages** define the stages that will be seen on the pipeline.
  Here, The pipeline will contain three stages: clean, compile and test.
- **Steps** define the commands that each stage will execute.
- **bat** is used here because the Windows Powershell will be used to execute Batch commands.
  If Linux is used, use **sh** instead of **bat**

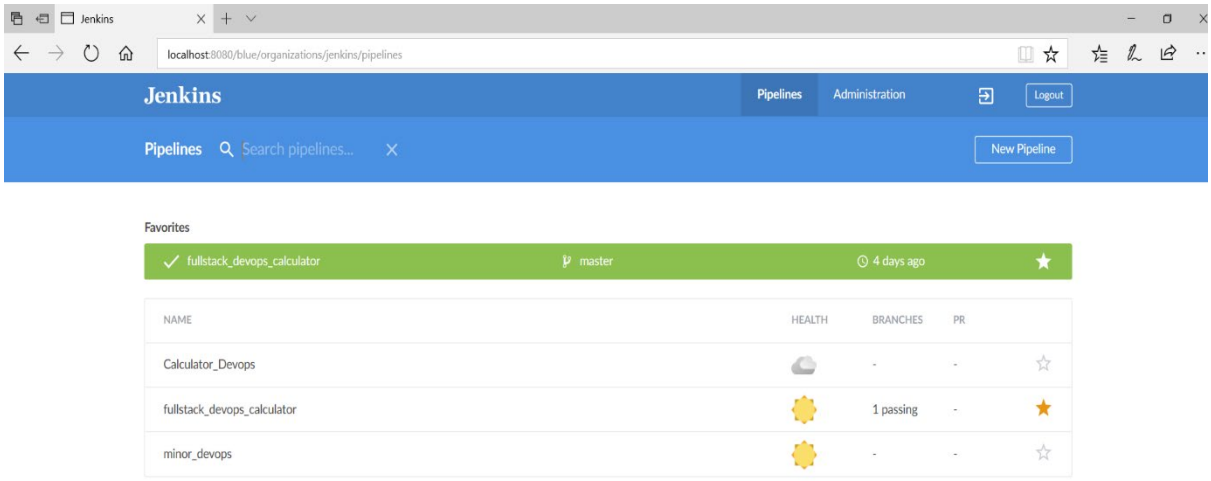
9.2 Install **BlueOcean Pipeline plugin** on Jenkins

Click on Manage Jenkins -> Manage Plugins -> BlueOcean Plugin



9.3 Click on **Open Blue Ocean** on the left sidebar of the Jenkins dashboard.

The BlueOcean dashboard opens up.

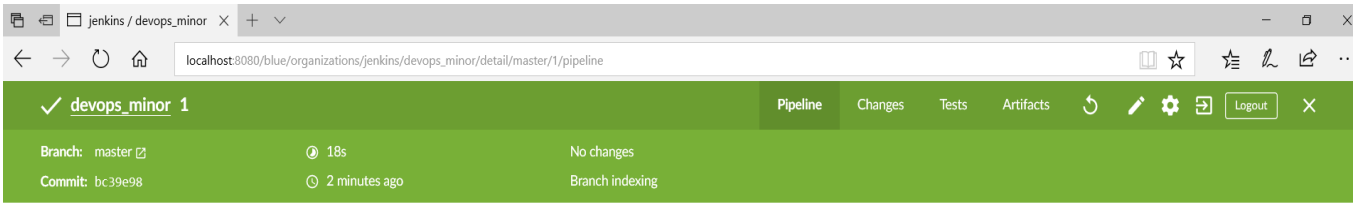


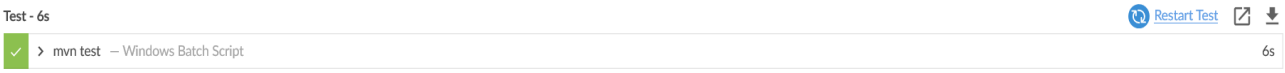
9.4 Click on **New Pipeline** on the top right.

- Specify the **Source Code Management** tool where the project is hosted. In this project, we use **GitHub**
- If your connected GitHub account is linked with multiple organizations, select the organization where your project is hosted. Here, the project belongs to **Rishika**
- Select the repository where your project is hosted.
  Here, the repository name is **minor_devops**
- Resolve naming conflicts, if any.

The pipeline is thus created and Jenkins automatically executes the pipeline whenever a change triggers it.

9.5 Click on the pipeline name to check its status.

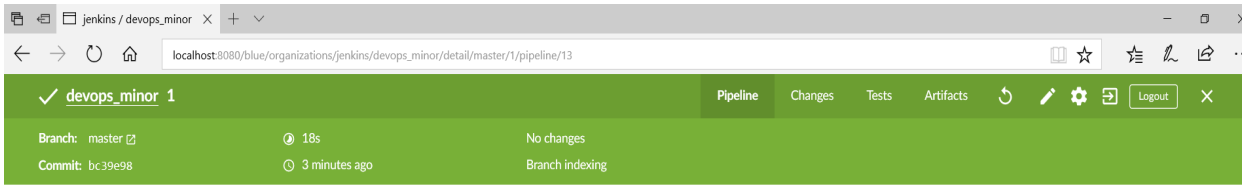


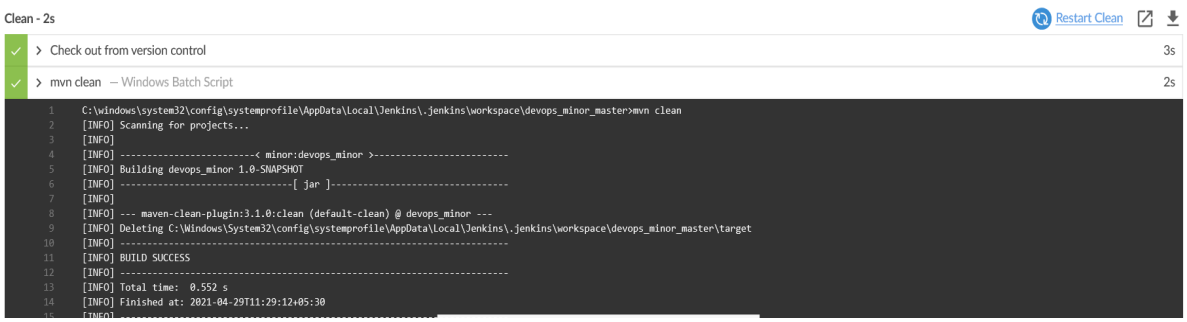


Here, the pipeline shows green color for all the stages which states that all the stages in the pipeline are up and running.

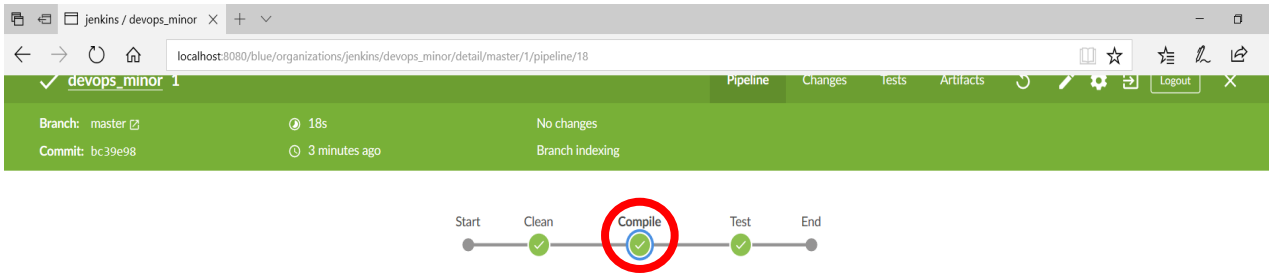
Click on individual stages to see the console output for each stage:

Clean

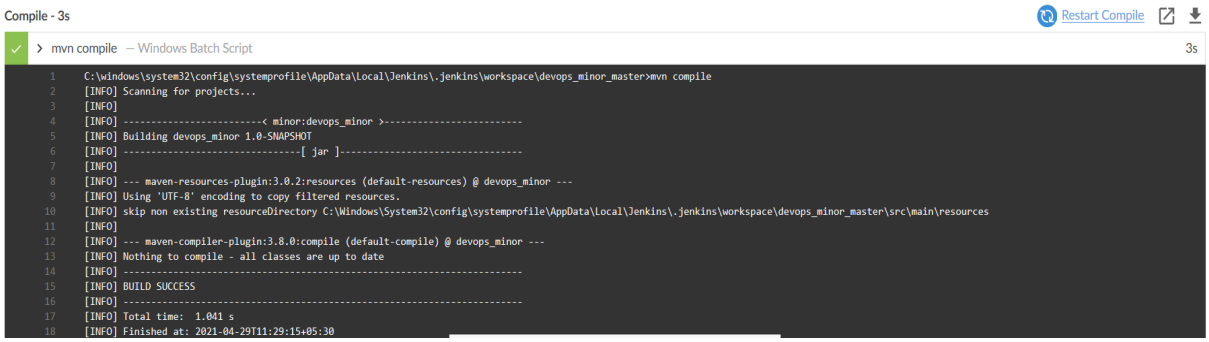




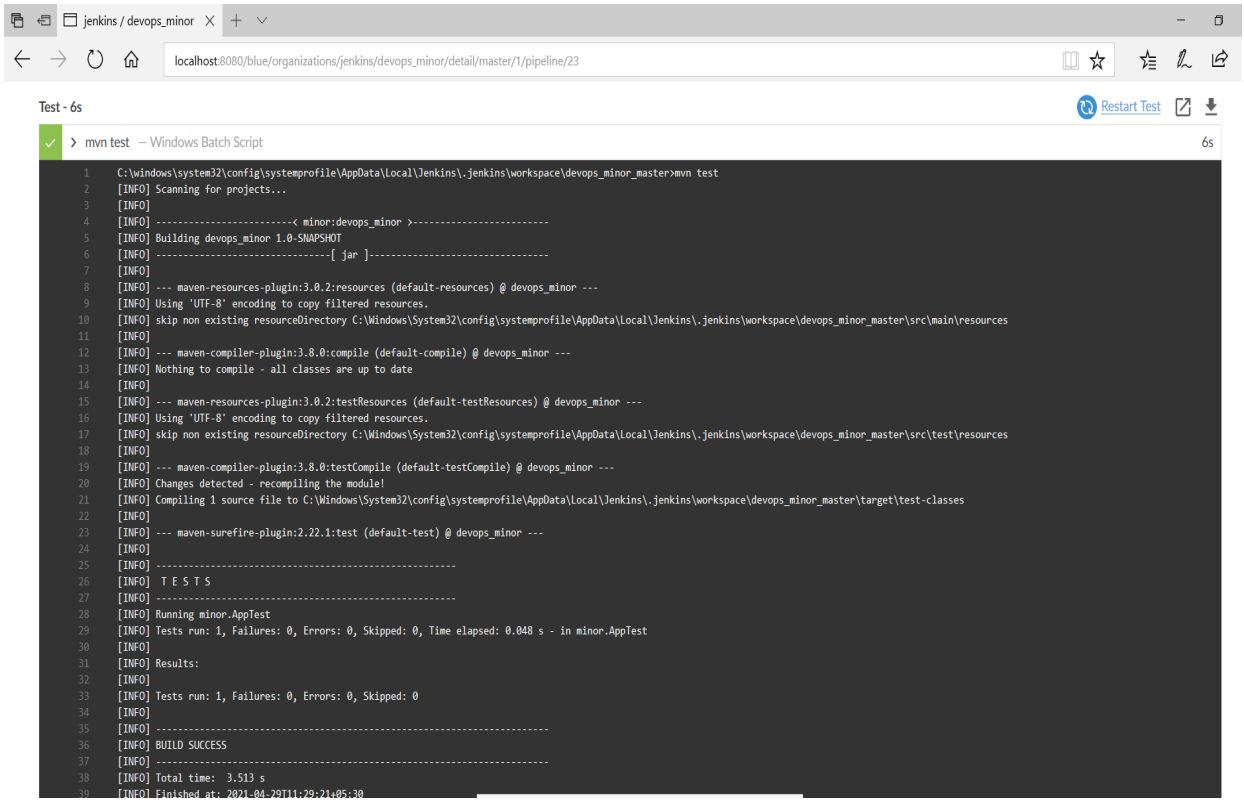
Compile


jenkins / devops_minor
localhost:8080/blue/organizations/jenkins/devops_minor/detail/master/1/pipeline/18
devops_minor 1
Pipeline
Changes
Tests
Artifacts
Logout
Branch: master
18s
No changes
Commit: bc39e98
3 minutes ago
Branch indexing

Start
Clean
Compile
Test
End


Compile - 3s
Restart Compile
mvn compile — Windows Batch Script
3s
C:\windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\devops_minor_master>mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< minor:devops_minor >------------------------
[INFO] Building devops_minor 1.0-SNAPSHOT
[INFO] --------------------------------[jar]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ devops_minor ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Windows\System32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\devops_minor_master\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ devops_minor ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 1.041 s
[INFO] Finished at: 2021-04-29T11:29:15+05:30

Test

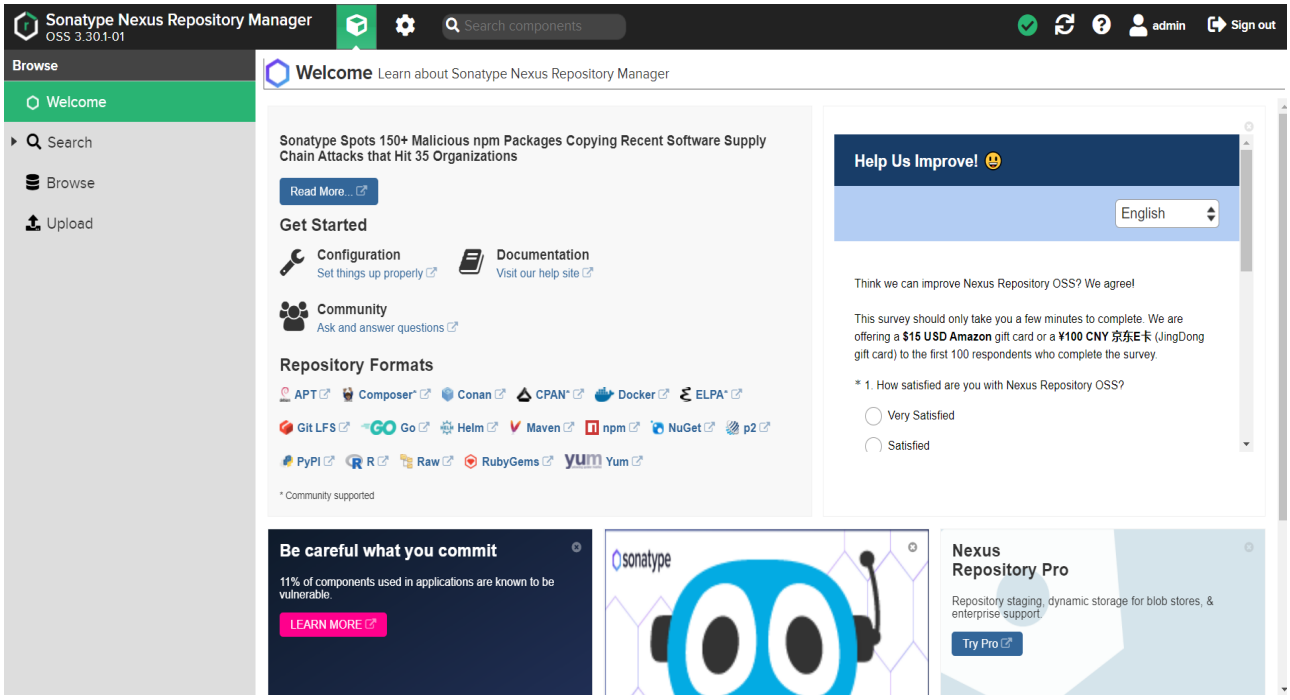

jenkins / devops_minor
localhost:8080/blue/organizations/jenkins/devops_minor/detail/master/1/pipeline/23
Test - 6s
Restart Test
mvn test — Windows Batch Script
6s
C:\windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\devops_minor_master>mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< minor:devops_minor >------------------------
[INFO] Building devops_minor 1.0-SNAPSHOT
[INFO] --------------------------------[jar]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ devops_minor ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Windows\System32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\devops_minor_master\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ devops_minor ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @ devops_minor ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Windows\System32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\devops_minor_master\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ devops_minor ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Windows\System32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\devops_minor_master\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ devops_minor ---
[INFO]
[INFO] ---
[INFO] T E S T S
[INFO] ---
[INFO] Running minor.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.048 s - in minor.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 3.513 s
[INFO] Finished at: 2021-04-29T11:29:21+05:30

5. Deploy the build artifacts to Nexus repository via Jenkins

Till now, Jenkins is used to automate the project work flow and execute maven goals on the project hosted on Git to create .jar files in the **target.** folder.

Open the Nexus Dashboard in a web browser at **localhost:8081** address.

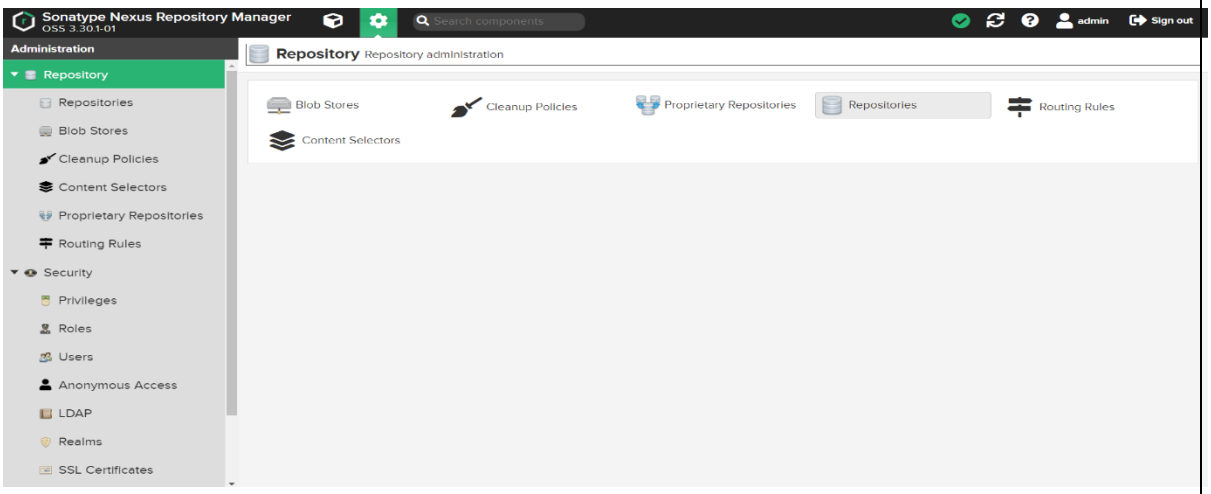
Nexus Dashboard opens up.



Create Repositories on Nexus

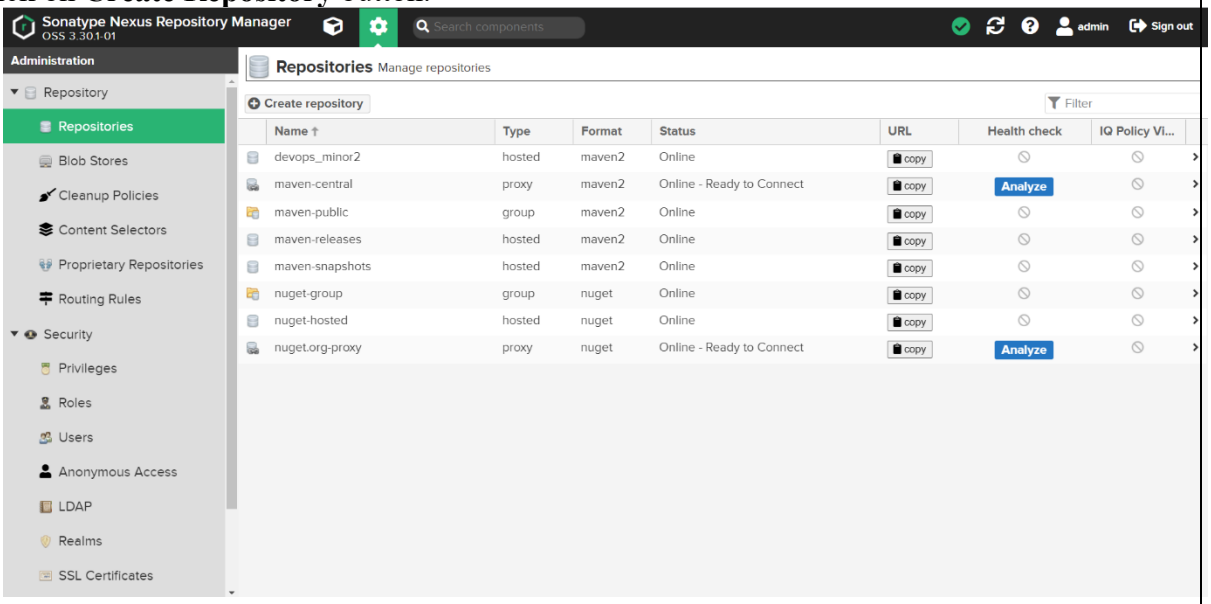
Create Maven compatible repositories on Nexus where our binary executable artifacts will be stored.

1. Click on Repository Settings button on the top bar of the Nexus Dashboard. The following dashboard opens up.

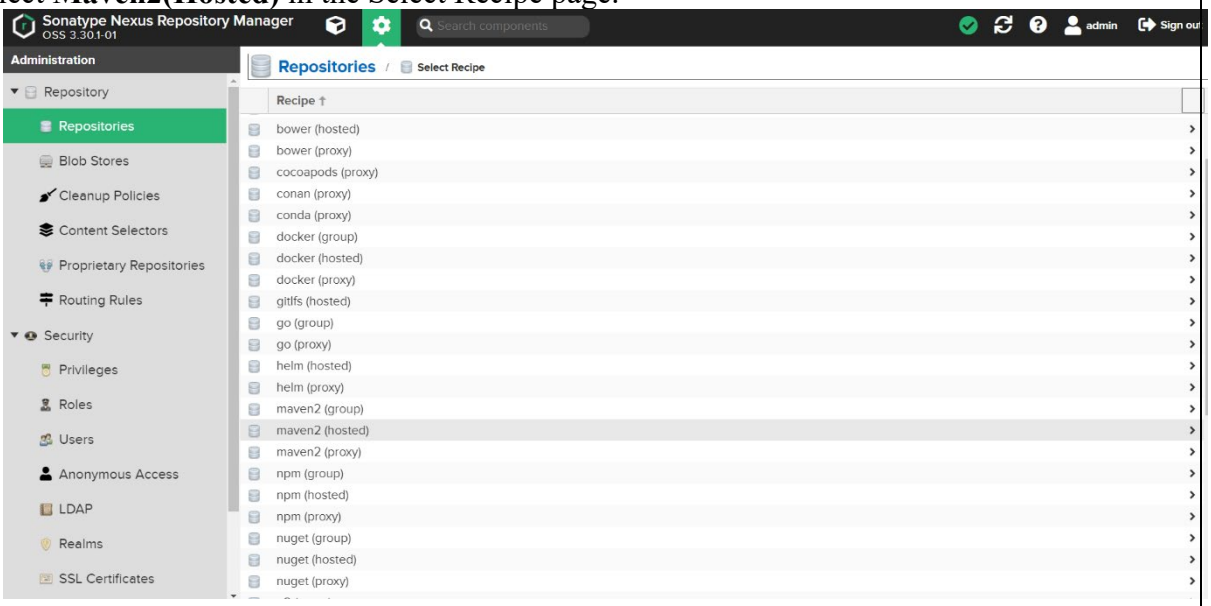


Click on Repositories.

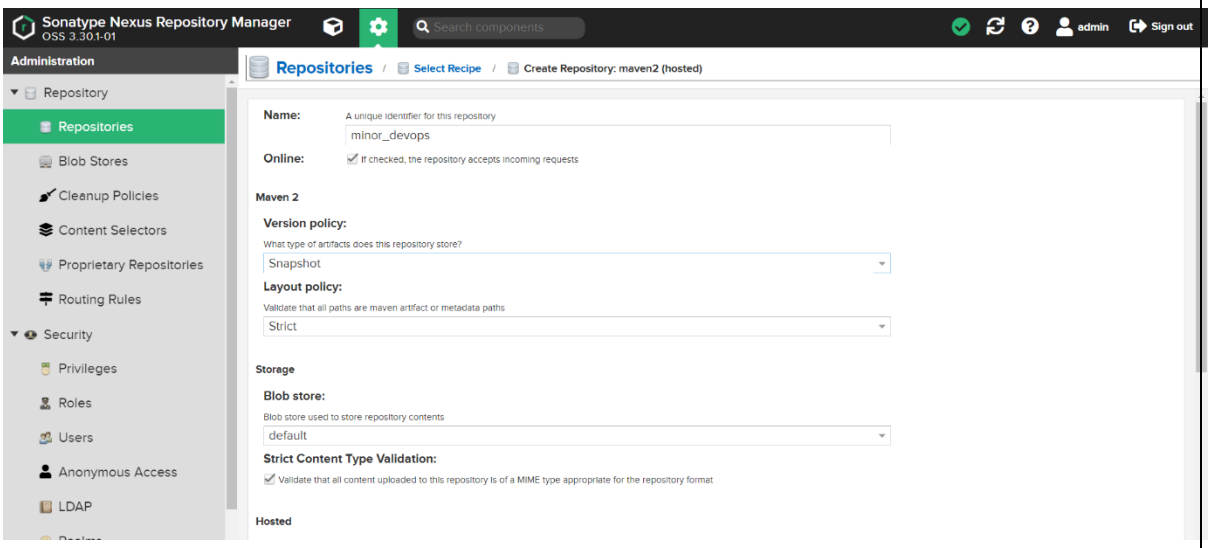
2. Click on **Create Repository** button.



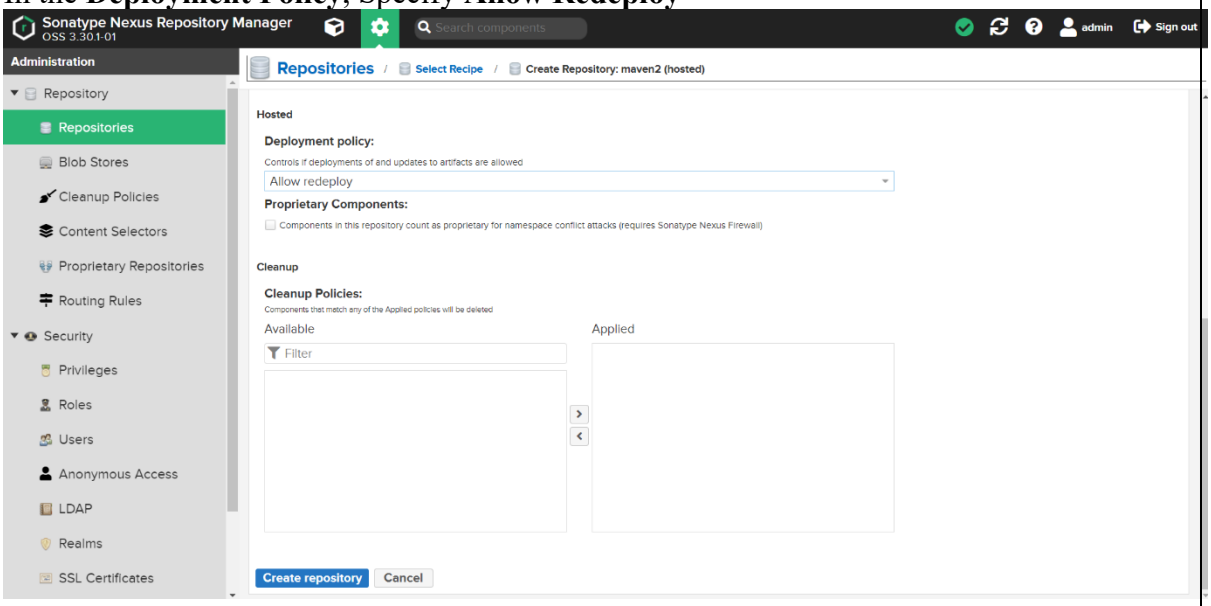
3. Select **Maven2(Hosted)** in the Select Recipe page.



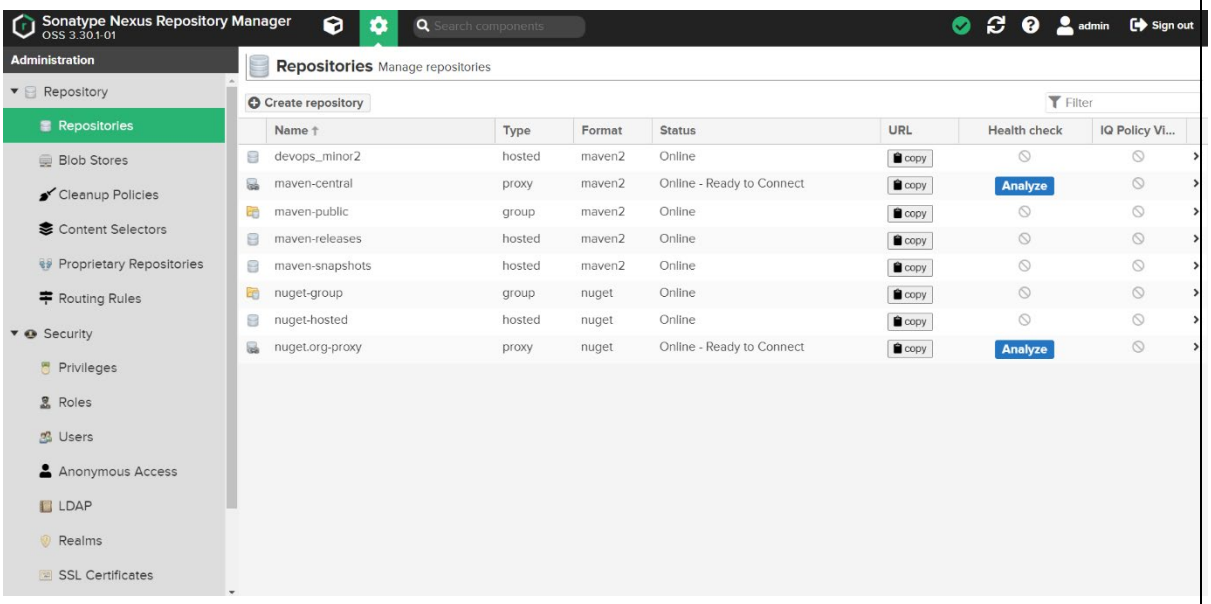
4. Specify the name of the repository. Here, **minor** is used as the artifact repository name. Also, specify the **Version policy** as Snapshot.



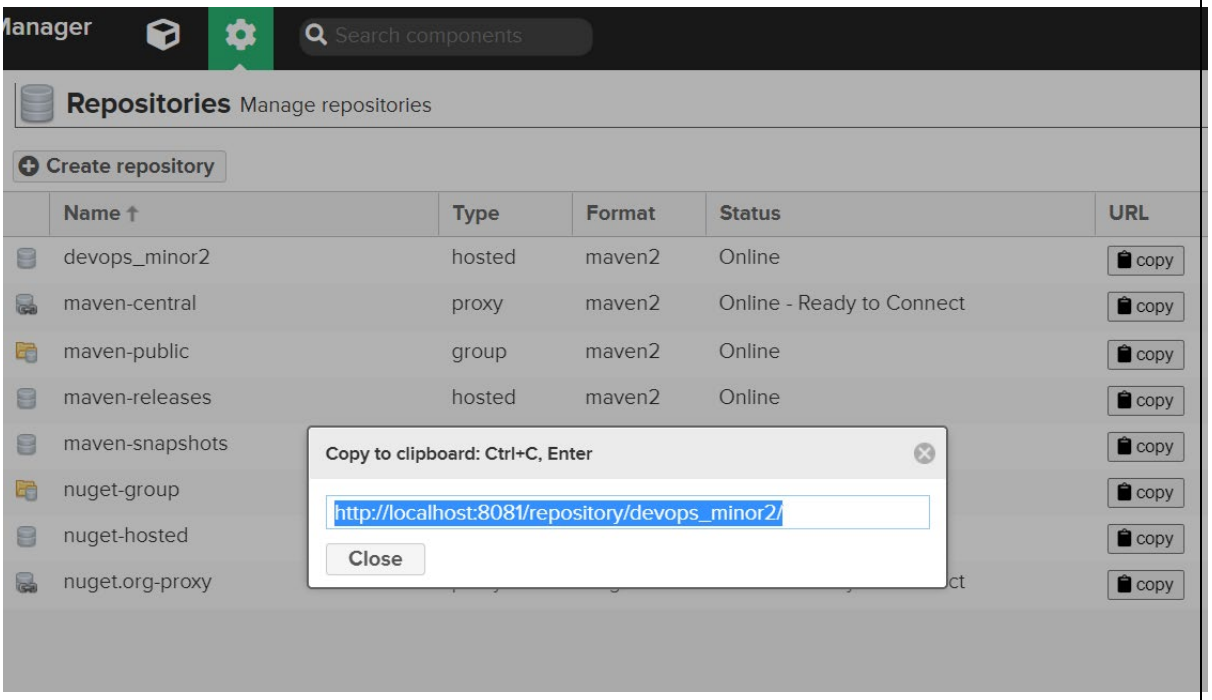
In the **Deployment Policy**, Specify **Allow Redeploy**



5. The repositories dashboard opens up with success message for repository creation.



Click on **Copy** button under URL tag for the devops_minor2 repository.



This link has to be configured with our Maven Settings on local machine and pom.xml file hosted on GitHub.

Configuring Repository link with Apache Maven settings on local machine

1. Go to {path of Apache Maven on local machine}/conf/
2. Open the settings.xml file
3. Go to <servers> tab and add following code

```
<server>
    <id>minor</id>
    <username>admin</username>
    <password>admin123</password>
</server>
```

Were,

- Id: the name of the repository created on the Nexus dashboard.
- Username: The name of the user having access rights to repository on Nexus. Here, admin user is used.
- Password: Authentication password of Nexus user specified above.

Save and close the settings.xml file

Configuring Repository link with POM.xml file hosted on GitHub
Edit the pom.xml file hosted on GitHub and add following code:

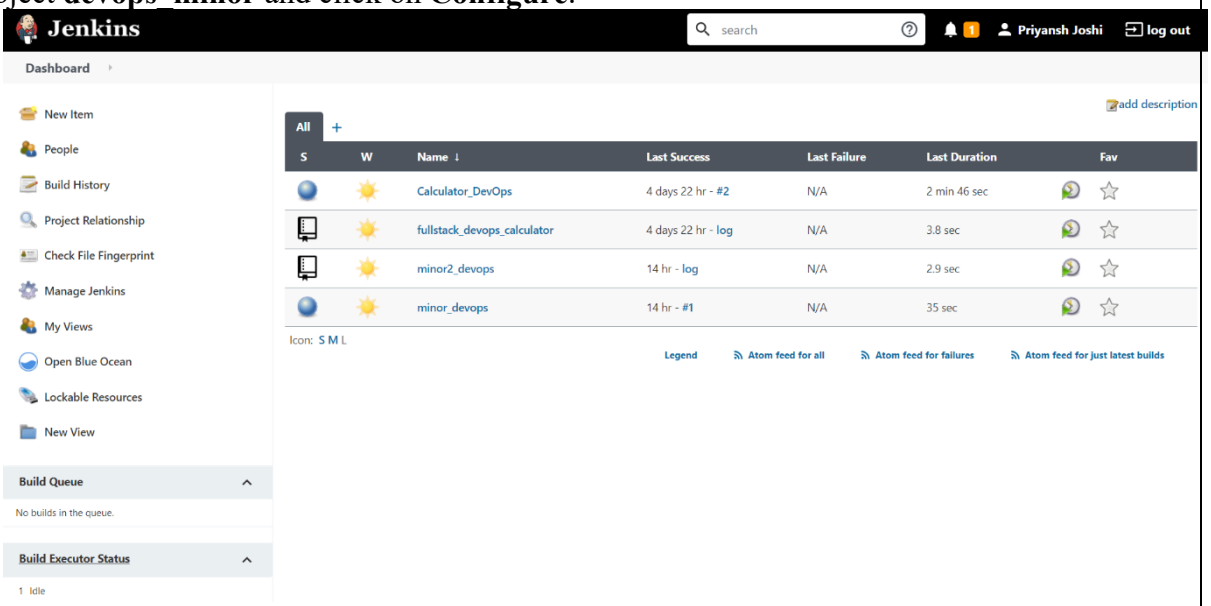
```
<distributionManagement>
  <snapshotRepository>
    <id>devops_minor</id>
    <name>minor</name>
    <url>http://localhost:8081/repository/devops_minor2/</url>
  </snapshotRepository>
</distributionManagement>
```

were,

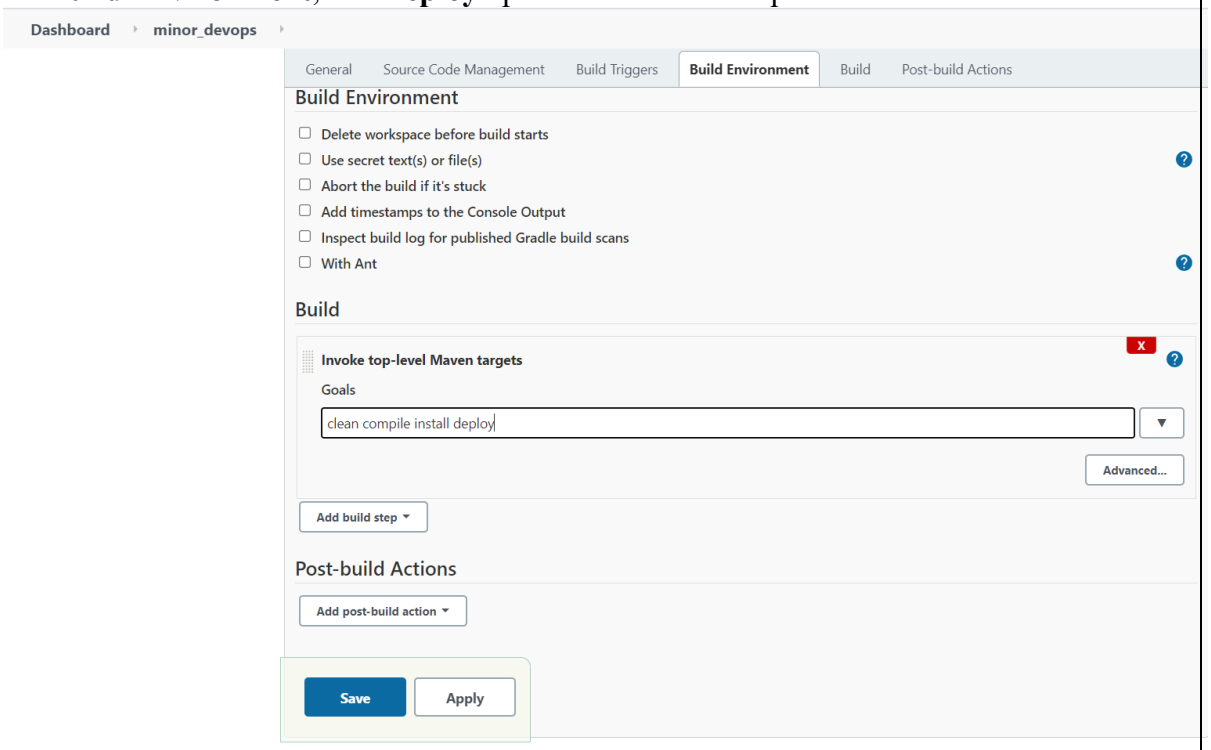
- Id: The name of the repository
- Name: The name of the repository
- URL: The link of the Nexus repository.

Automate the deployment of our binaries using Jenkins

1. Go to Jenkins Dashboard and right click on the job created for current project **devops_minor** and click on **Configure**.



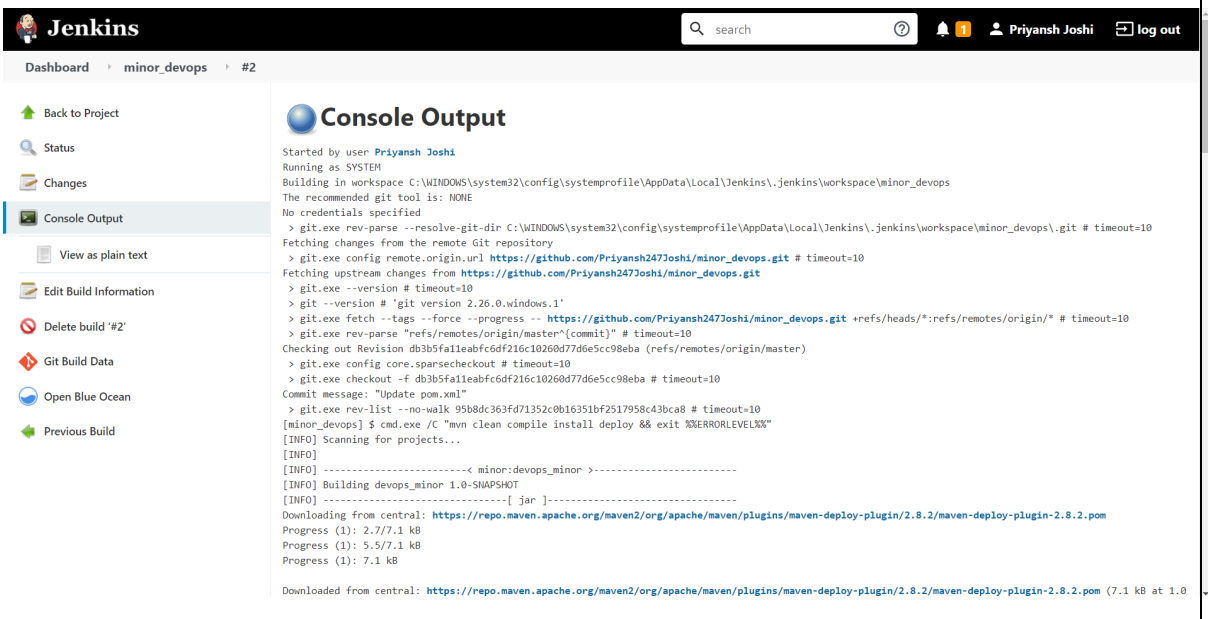
2. In the **Build Environment**, add **Deploy** option in the Build steps.



Click on Save.

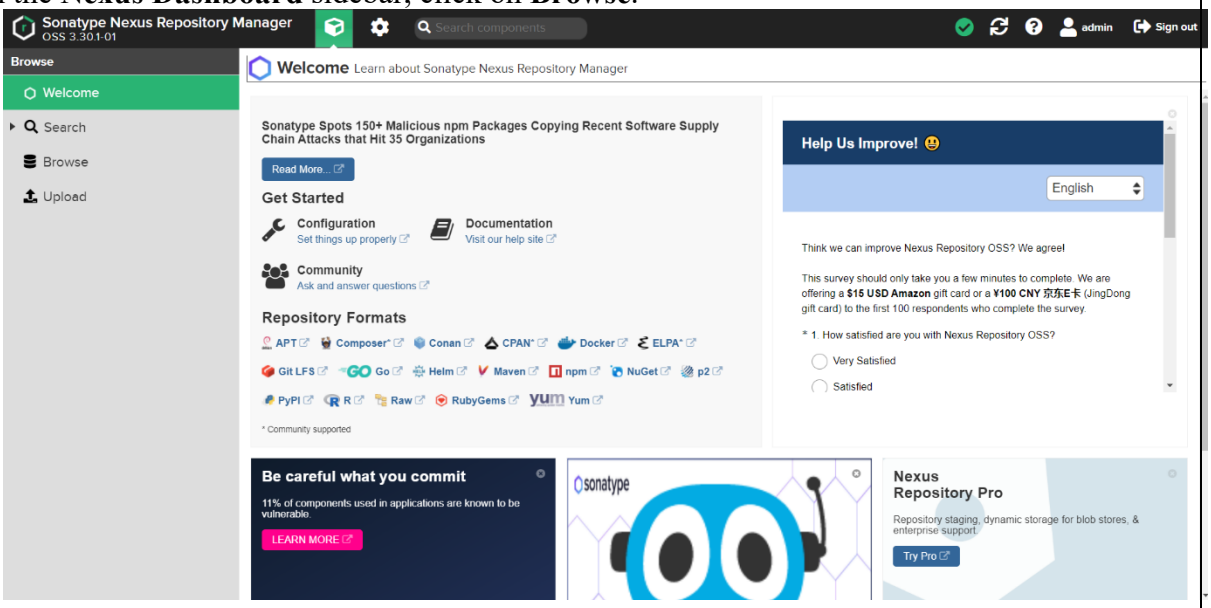
3. On the project dashboard, click on **Build Now**

4. See the console output for the Build

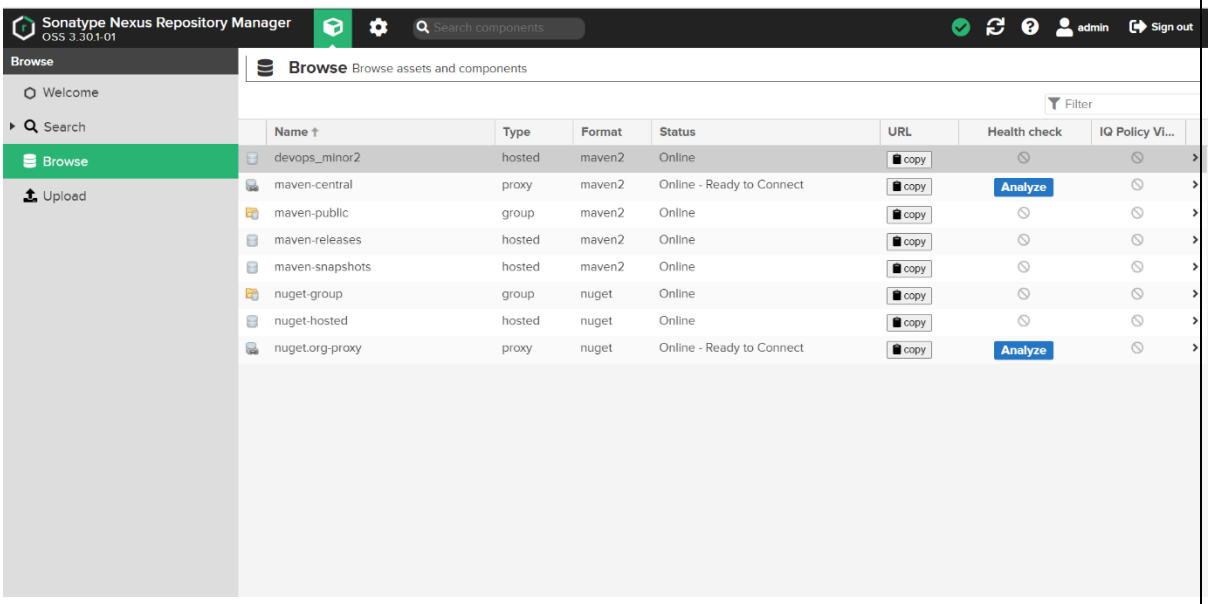


Verify if the binaries are deployed on Nexus

1. On the **Nexus Dashboard** sidebar, click on **Browse**.



2. Select the **devops_minor2** repository to check its content.



3. The below image shows that the binaries are successfully deployed on Nexus.